Kaggle 個人競賽實驗報告 Kaggle Individual Competition Report

張哲安/M946113003/大數據科技及管理研究所

一、資料集前處理 Dataset Preprocessing

在文本資料中發現有相同文本對應多種不同標籤的問題，共 4999 筆相同文本，7995 筆沒有重複的文本。嘗試改變任務為多標籤分類，並在 Kaggle 資料預測上，皆輸出 Logit 最高的，作為最終標籤的輸出。另外因為重複的文本皆有著不同且不重複的標籤，Kaggle 有著 46 筆重複文本，與訓練有 669 筆重複。訓練模型的資料如果包含這些可能為雜訊，造成預測錯誤。故先切分好訓練驗證資料，保持驗證資料與 Kaggle 有著類似的資料狀況，刪除訓練資料重複的資料，再進行模型訓練，這是假設 Kaggle 的數據分佈狀況跟訓練資料一樣的話，那標籤不重複，模型一定無法預測出正確的結果。除此之外也嘗試隨機刪除一筆以及按照類別比例留取一筆。在使用一般機器學習，轉換數字向量前，進行還原字、刪除 stop word、將所有字改為小寫、刪除過多的空格。而如果是使用 BERT 模型，雖然有嘗試進行還原字、刪除 stop word、將所有字改為小寫、刪除過多的空格，但並不會提升模型表現，故不做前處理盡量不改變原文原意。對於不平衡資料問題上，嘗試進行數據增強，包含同義詞替換、上下文詞替換、翻中文在翻回英文及 Gemini 改寫，只針對少數類樣本進行增強，先以每一筆資料做一次的增強，如資料量仍無法達成平衡，隨機抽取資料做增強至數量與最多類樣本數量一致，不但可以變為平衡資料，並且還可以增加文本的多樣性，提升模型表現及泛化能力。

二、系統架構及方法說明

對於不平衡資料上的處理，嘗試同時使用 class weight 及 focal loss，讓模型同時增加少數類樣本的權重，及減少對多數類樣本的關注。在切 token 及轉為數字向量上，使用 spaCy 醫療自然語言的套件進行斷詞，並使用 TF-IDF 轉為數字向量。另外嘗試使用 gemini 所開發的 text embedding gecko。轉換成數字向量後，再進行後續常見的機器學習方式，以及進行數字向量標準化前後的模型訓練，包含 naïve bayes、logistic regression、decision tree、knn、svm 及 xgboost。在使用 Bert 模型時，因為運算資源得受限，使用動態 padding 的方式，先根據 token 的長度進行由小至大的排序，再 Padding 至每一個 Batch 中最長的句子。最長至 512Token，超過則截斷。因為想要進一步提升單一 BERT 模型的成績，先嘗試 Voting 的方式，使用單一 Biobert、Biolinkbert、Clinical Modernbert 進行投票。另外也嘗試使用 Ensemble soft voting 的想法，選擇 Bio_Clinical Bert、scibert 以及 Bio_Clinical Bert、gatortron，使用不同專業領域資料進行微調的模型，以及不同層數的 BERT。在對相同文本時，可以捕捉學習到不同的特徵資訊，達到互補的效果。但因兩個 BERT 模型不會表現一樣好，我也不知道哪一個表現較好，故把兩個 BERT 輸出的 5 類 logit 值進行加權總合，並且權重由訓練模型時，由模型自行訓練而得，並且因模型架構變複雜，固為預防過擬合，在加權總合後加入 Dropout 後再進行 5 類分類。在參數設定上，相同的設定為，Batch size 設定為 2，並使用累積梯度的方式累積至 16 再更新計算權重。Optimizer

選擇使用 adamw torch fused，相比於一般 adamw，可以較好的整合 CUDA，提升運算效率。Learning rate 設 2e-5，Warmup ratio 設 0.1，並且做線性遞減。總共訓練 10 個 Epochs，設定 Early stopping 為 2，選擇驗證集 F1 macro 較高的模型，使用混合精度來計算 weight 及 bias，以及 Dropout 設定 0.1。在實驗紀錄中，機器學習方法包含 TF-IDF、數字向量標準化及 Gecko 三種方式。自行使用的方法，呈現資料前處理前後，以及 Voting、Esemble 使用不同組合的 BERT。

三、實驗記錄 Experiments

1.Machine Learning Classifiers

Method	Training Data k-fold Average-Micro Precision/Recall/F1	Training Data k-fold Average-Macro Precision/Recall/F1	Test Data Public Leaderboard Score/Rank/filename	Test Data Private Leaderboard Score/Rank/filename	Parameters for training model
Naïve Bayes#1	0.585/0.585/0.585	0.63/0.506/0.516	0.503/NB0520	0.487	NA
Naïve Bayes#2	0.643/0.643/0.643	0.631/0.698/0.646	0.612/NB1_0520	0.675	NA
Naïve Bayes#3	0.643/0.643/0.643	0.631/0.698/0.646	0.612/NB_0522	0.675	NA
Logistic Regression #1	0.643/0.643/0.643	0.654/0.724/0.664	0.629/LR0501	0.686	class_weight=balanced,multi_class=multinomial
Logistic Regression #2	0.635/0.635/0.635	0.645/0.716/0.655	0.578/LR0502	0.631	同上
Logistic Regression #3	0.61/0.61/0.61	0.596/0.648/0.61	0.591/LG0519	0.624	同上
Decision Tree #1	0.411/0.411/0.411	0.4/0.418/0.407	0.411/DT_0522	0.395	class_weight=balanced
Decision Tree #2	0.326/0.326/0.326	0.321/0.328/0.324	0.333/DT1_0522	0.333	同上
Decision Tree #3	0.327/0.327/0.327	0.322/0.329/0.325	0.333/DT2_0522	0.328	同上
KNN #1	0.541/0.541/0.541	0.523/0.538/0.526	0.528/KNN0519	0.508	NA
KNN #2	0.596/0.596/0.596	0.58/0.611/0.59	0.613/KNN0520	0.594	NA
KNN #3	0.593/0.593/0.593	0.577/0.608/0.587	0.604/KNN1_0520	0.606	NA

SVM #1	0.836/0.836/0.836	0.822/0.884/0.838	0.293/svc2_0519	0.297	kernel=poly, class_weight=balanced
SVM #2	0.585/0.585/0.585	0.566/0.603/0.579	0.576/svc1_0519	0.615	同上
SVM #3	0.631/0.631/0.631	0.619/0.681/0.629	0.609/svc0519	0.665	同上

2.你的方法 Your Methods

Method	Training Data k-fold Average-Micro Precision/Recall/F1	Training Data k-fold Average-Macro Precision/Recall/F1	Test Data Public Leaderboard Score/Rank/filename	Test Data Private Leaderboard Score/Rank/filename	Parameters for training model
XGBoost#1	0.728/0.728/0.728	0.749/0.692/0.71	0.519/2/XGB0328	0.491	加 sample_weights
XGBoost#2	0.739/0.739/0.739	0.755/0.697/0.717	0.535/2/XGB1_0328	0.517	同上
Biobert#1	0.654/0.654/0.654	0.64/0.698/0.653	0.639/3/biobert0424	0.672	不同設定 gradient_accumulation_steps=4, epochs=1
BioLinkBert#1	0.705/0.705/0.705	0.697/0.726/0.702	0.624/2/BioLinkBERTBASE0401	0.666	同上
BioLinkBert#2	0.71/0.71/0.71	0.7/0.729/0.708	0.627/2/BioLinkBERTBASE0406	0.679	同上
Clinical ModernBert#1	0.661/0.661/0.661	0.649/0.692/0.66	0.622/2/CLIMODBERT0419_2	0.656	同上
Clinical ModernBert#2	0.668/0.668/0.668	0.654/0.696/0.666	0.638/2/CLIMODBERT0420	0.645	同上
BioClinicalBert#1	0.721/0.721/0.721	0.711/0.735/0.72	0.622/2/BioClinicalBERT0401	0.68	同上
BioClinicalBert#2	0.734/0.734/0.734	0.708/0.692/0.7	0.668/1/BioClinicalBERT0507	0.711	同上
Vote#1	0.659/0.659/0.659	0.634/0.7/0.656	0.642/3/vote1_0425	0.676	見系統架構及方法說明

Weightedesemble#1	0.715/0.715/0.715	0.7/0.743/0.713	0.71/1/weightedesemble2_0507	0.728	同上
Weightedesemble#2	0.728/0.728/0.728	0.711/0.734/0.722	0.718/1/new0527	0.74	同上

4.討論 Discussion

在轉換數字向量上，當可以更好的表示文本時，所得到的結果也就越好，這也是為什麼 Gecko 轉換後訓練的模型可以比使用 TF-IDF 來的好。在相同文本不同標籤的處理上，改為多標籤並無法幫助模型提升表現，可能是因為無法知道重複文本順序性。另外在進行數據增強後，雖然對模型表現上有提升，但因數據量增加造成運算成本更高，結果跟選擇使用訓練模型時加權重的方式有著差不多的表現，故最後選擇使用加權重之方式。資料的前處理仍然是相當重要的，當移除的相同的文本資料，讓訓練模型時減少雜訊的干擾，模型的表現有著明顯顯著的提升。另外在 BERT 模型選擇上，雖然也有可能是我訓練模型有所差別，但我發現使用與自己資料任務更相近的 Domain 資料微調過的模型，比使用更先進的框架的 BERT 來的更重要，例如過程中我發現單一個模型，我的 Biobert 可以比 Scibert 好，又可以比 BioCliniclBert 好，這顯示在相同架構下，相較於臨床文本，生物科學醫學文獻更貼近我的任務文本。比較使用更新的 BERT 架構並且是有使用臨床文本微調過的，但 Biobert 及 Scibert 仍然可以做得比 Clinical ModernBert 來的好。並且在更新今年剛推出的 Neobert，但並未使用任何生物科學醫學臨床文本微調過的模型，上述的模型表現皆可以比此表現來的好。在嘗試 Voting 上，我發現當把每一個模型視為同等重要一票時，時常發生最終成績會被拉低，更請向於表現較差的模型。另外還會發生不管是使用單數或雙數個模型，都會出現相同，每一個模型給出不一樣的答案或兩種答案出現的次數相同的問題。在把 BERT 進行 Ensemble 方法上，嘗試單存把 Logit 進行合併或平均時發現，效果並不是很好，我個人認為是因為跟 Voting 一樣的原因，把兩個模型視為相同重要進行，很容易會被表現較差的單一 BERT 拉下成績。所以不但需要有兩個 BERT 有者差不多的成績，但在分類結果上要有所差異，並且使用加權總合的方式，通常理論上效果可以比等權重及 Voting 來的好。另外也嘗試在 Ensemble 時，增加更多的模型，以及使用 CLS 進行平均或合併時，因為特徵維度資訊變大，當直接分 5 類時，模型訓練成績較差，可能因資訊丟失的狀況，固嘗試增加全連階層，加深類神經網路，去捕捉更多更複雜的特徵資訊，但也因此模型架構變得更為複雜，增加模型亦是如此，變為較難訓練，最終模型表現較差，上傳的成績也會 Overfitting 較嚴重。另外在過程中發現，模型在每類的表現上，第 5 類都會有較差的表現，故我嘗試把任務分為兩階段，第一階段單存做一個二元分類，是第 5 類及不是第 5 類，在第二階段中，把第 5 類排除，單存做 4 類的分類，但在最終模型的表現我發現很大程度是取決於第一階段的模型表現，如果第一階段無法很顯著的提升表現，最終的模型表現，也只會比第一階段來的更差。

5.外部資源與參考文獻

[1] C. McCormick, "Smart Batching Tutorial - Speed Up BERT Training," McCormickML, Jul. 29, 2020. [Online]. Available: https://mccormickml.com/2020/07/29/smart-batching-tutorial/

[2] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune BERT for text classification?" in *Proc. China National Conf. on Chinese Computational Linguistics*, Cham: Springer, 2019, pp. 194–206.

[3] Dataroots, "Fine-tuning BERT for an unbalanced multi-class classification problem," Dataroots Blog. [Online]. Available: https://dataroots.io/blog/incident-team-prediction

[4] E. Alsentzer, J. R. Murphy, W. Boag, W. H. Weng, D. Jin, T. Naumann, and M. McDermott, "Publicly available clinical BERT embeddings," *arXiv preprint arXiv:1904.03323*, 2019.

[5] Hugging Face, "Optimizer schedules," Hugging Face Transformers Documentation. [Online]. Available: https://huggingface.co/docs/transformers/v4.52.2/en/main_classes/optimizer_schedules#transformers.SchedulerType

[6] Hugging Face, "Trainer class," Hugging Face Transformers Documentation. [Online]. Available: https://huggingface.co/docs/transformers/main_classes/trainer

[7] Hugging Face, "TrainingArguments," Hugging Face Transformers Documentation. [Online]. Available: https://huggingface.co/docs/transformers/main_classes/trainer#transformers.TrainingArguments

[8] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A pretrained language model for scientific text," *arXiv preprint arXiv:1903.10676*, 2019.

[9] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, Feb. 2020. doi: 10.1093/bioinformatics/btz682.

[10] L. L. Breton, Q. Fournier, M. E. Mezouar, and S. Chandar, "NeoBERT: A Next-Generation BERT," *arXiv preprint arXiv:2502.19587*, 2025.